

## Introducere

GPSS (General Purpose System Simulation) este un limbaj pentru simularea modelelor. Orice sistem poate fi descris prin intermediul unui număr determinat de elemente standard – așa zise obiecte. Regulile logice care stau la baza sistemului, poate fi redus la un set de operații simple. Corespunzător un limbaj de modelare trebuie să fie alcătuit din obiecte și operații abstracte.

Categorii și tipuri de obiecte ale GPSS.

Categoria	Tipul
Dinamică	Tranzacții
Operațională	Blocuri
Aparataj	Dispozitive, Echipamente multicanal, Chei logice
Calcul	Variabile (aritmetice, logice), Funcții
Statistică	Șiruri de așteptare (cozi), Tabele
Memorie	Celule de memorie, Matrice de celule
Grupare	Lanțuri (liste), Grupuri

**Tranzacțiile** sunt obiecte ale categoriei *dinamice*. Acestea sunt părți componente ale fluxurilor cercetate, care parcurg (se mișcă) structura fixată a modelului (ce constă din obiecte ale altor categorii) și care îndeplinesc un șir de operații indicate.

Obiectele *operaționale* (**blocurile**) redau logica e funcționare a modelului și determină calea de parcurgere a tranzacțiilor prin obiectele categoriei de *aparataj*.

Obiectele categoriei de *aparataj* sunt elemente abstracte prin intermediul cărora este alcătuit modelul sistemului real. Acționând asupra acestor obiecte, tranzacțiile pot să schimbe starea lor, ori să influențeze la parcurgerea altor tranzacții.

Categoria de *calcul* este destinată pentru descrierea situațiilor, când legătura dintre componentele sistemului modelat poate fi simplu descrisă cu ajutorul expresiilor matematice.

În scopul evaluării caracteristicilor ale sistemului modelat se folosesc registratori ca **șiruri de așteptare** și **tabele**. (*Categoria de statistică*). Ele nu influențează la funcționarea modelului, destinate numai pentru culegerea informației statistice. Însă indirect (cu parametrii lor), ele pot să influențeze la funcționarea modelului.

Categoria de *memorie* este destinată pentru înscrierea valorilor necesare în **celule** și **matrice de celule**.

GPSS permite, în caz de necesitate, de a include tranzacțiile în **liste** și **grupuri**. De asemenea, este posibilă îndeplinirea operațiilor de grup asupra tranzacțiilor din liste.

Pentru descrierea structurii modelării sistemelor deseori se folosește prezentarea lor grafică prin scheme-bloc. Ele constau dintr-un număr determinat de blocuri, fiecare din ele prezentând o etapă de funcționare a modelului, corespunzător cu logica lui de funcționare. Blocurile sunt unite prin linii, care indică consecutivitatea de parcurgere a tranzacțiilor. Ele pot prezenta și evenimentele ce se petrec în sistem.

Aproape fiecare bloc are un număr determinat de intrări și la fel aproape fiecare are un număr determinat de ieșiri. Alegerea căii de parcurgere la ieșirea din bloc este sau aleator, sau după un algoritm oarecare. Tranzacția poate să se afle numai în bloc. Acțiunea, descrisă în bloc se îndeplinește numai la intrare tranzacției în acest bloc. Blocul poate să respingă intrarea tranzacției în urma unor condiții de blocare. În GPSS fiecare bloc are configurarea sa specială. În așa mod bloc-schemele sunt unificate.

Tranzacțiile în GPSS (versiune DOS) pot avea până la 100 parametri și pot avea de asemenea un nivel de prioritate (128 nivele de prioritate). Fiecare tranzacție are numărul său de identificare (acest identificator nu este accesibil pentru utilizator ca și timpul de generare a tranzacției). În procesul de modelare parametrii și prioritatea pot fi unificate și valorile lor pot fi folosite de obiectele categoriei de *calcul* și de asemenea de blocurile ce schimbă direcția de parcurgere a tranzacțiilor.

Pentru a modela un sistem, el trebuie să fie descris în termeni GPSS. Programul ce dirijează cu: parcurgerea tranzacțiilor, schimbarea stărilor obiectelor, culegerea statisticii se numește **simulator**. După lansarea programului la execuție, simulatorul generează tranzacții, exercită parcurgerea lor prin blocuri, îndeplinește acțiunile respective, indicate în blocuri. (Tranzacțiile se mișcă din bloc în bloc asemenea unor fluxuri). Orice schimbare a stării sistemului se numește eveniment. Simulatorul fixează timpul de apariție a evenimentelor, prelucrându-le după acest timp.

Pentru menținerea corectă a consecutivității evenimentelor în timp, în sistemul modelat funcționează **ceasul de sistem**, care indică timpul de modelare. Indicațiile ceasului în orice moment de timp se numește timpul absolut al modelului. Timpul absolut nu poate fi schimbat de către utilizator. Există la fel și timp relativ, care poate fi schimbat cu ajutorul unor cartele speciale.

GPSS funcționează numai cu numere întregi și corespunzător timpul modelului poate primi numai valori întregi.

Unitatea de măsură a timpului (sec., min., oră, an ș.a.) se alege de către utilizator. Simulatorul însă lucrează cu unități convenționale (timpul modelului). Simulatorul nu analizează starea sistemului în fiecare unitate de timp, dar schimbă valoarea timpului absolut, egalându-l cu timpul celui mai apropiat eveniment. Corespunzător timpul de modelare nu depinde de intervalul de timp în sistemul real, dar depinde de numărul de evenimente prelucrate.

## Lanțuri în GPSS

În GPSS tranzacțiile pot să se afle în câteva lanțuri (lanțurile sunt echivalente cu listele). Cu toate acestea există 5 tipuri de lanțuri de evenimente:

1. Lanțul evenimentelor curente
2. Lanțul evenimentelor viitoare
3. Lanțul întreruperilor
4. Lanțul tranzacțiilor în pereche
5. Lanțul utilizatorului

În lanțul evenimentelor curente se află tranzacțiile cu timpul evenimentului egal cu timpul modelului. Deservirea tranzacțiilor în lanțul evenimentelor curente se îndeplinește conform priorității tranzacțiilor (cu cât este mai mare prioritatea, cu atât mai înainte ea se deservește). Tranzacțiile din aceeași clasă (cu aceeași prioritate) sunt distribuite conform timpului de sosire a tranzacției (cu cât mai devreme a sosit cu atât mai aproape de începutul cozii din clasa sa ea stă).

La deservirea tranzacției simulatorul o mișcă înainte prin blocuri cât este posibil. La nimerirea ei într-un bloc cu condiții de blocare sau într-un bloc cu reținere de timp, parcurgerea ei încetează, iar simulatorul deservește următoarea tranzacție.

În lanțul evenimentelor viitoare sunt incluse acele tranzacții, ale căror timp de declanșare este mai mare decât timpul curent al modelului. Ordinea de aranjare a tranzacțiilor în acest lanț este conform timpului de declanșare (prioritatea nu se ia în considerație). Din lanțul evenimentelor viitoare, tranzacțiile trec în lanțul evenimentelor curente, când le vine timpul.

Lanțul utilizatorului conține acele tranzacții, care sunt extrase din model de către utilizator.

În lanțul de întreruperi se află acele tranzacții, deservirea cărora a fost întreruptă. Aceste tranzacții se includ în model după anularea întreruperii.

Lanțul tranzacțiilor în pereche conține acele tranzacții, ale căror parcurgere de mai departe este imposibilă din lipsa unei sau a mai multe tranzacții (în dependență de blocul utilizat).

## Sintaxa limbajului GPSS

Pentru a scrie un program în GPSS se folosesc literele latine [A..Z] și numerele [0..9], de asemenea și simboluri speciale. Este de dorit ca programul să fie scris într-o format standard, care constă din 3 câmpuri:

Câmpul	Poziția
Etichetelor	2..6
Blocurilor	8..18
Operanzilor	19..73

Câmpul operanzilor permite descrierea până la 8 operanzi. Poziția operanzilor este notată cu litere latine: A,B,C,D,E,F,G,H.

Operanzii se despart între ei prin virgulă (întrebuințarea spațiilor în acest scop este interzisă). La absența unui operand și prezența informației după el, acesta este identificat prin virgulă.

ABC GENERATE 100,,3

Dacă după operand nu este o altă informație virgula se omite.

ABC GENERATE 100

De obicei GPSS folosește numere întregi. Formatul numerelor poate fi cuvânt (așa număr se păstrează în 4 octeți [ $2^{31}-1$ ]) și semicuvânt (2 octeți [ $2^{15}-1$ ]).

Orice informație în câmpul operanzilor după spațiu este primită ca comentariu. Pentru înscrierea comentariilor lungi este necesar ca în prima poziție a rândului să fie înscris semnul “\*”. În acest caz rândul în întregime este socotit ca comentariu și este ignorat de către simulator.

### Atribute numerice standard (ANS)

În GPSS fiecare obiect dispune de un șir de atribute, care descriu starea lui în momentul dat. Atributele pot avea valori logice sau numerice. Atributele, care sunt accesibile de utilizator se numesc standarde.

Tipul obiectului	SNA	Diapazonul	Descrierea
Tranzacția	$P_j$	$2^{31}-1$	Valoarea parametrului $j$
	PR	0..127	Nivelul de prioritate
	M1	$2^{31}-1$	Timpul tranzacției [C1-T <sub>GE</sub> ]
	MP <sub>j</sub>	$2^{31}-1$	Timpul relativ al tranzacției [C1-P <sub>j</sub> ]
Blocul	$N_j$	$2^{24}-1$	Numărul de intrări
	$W_j$	$2^{15}-1$	Conținutul curent
Dispozitivul	$F_j$	0 sau 1	Starea: 0 – liber; 1 – ocupat
	FC <sub>j</sub>	$2^{31}-1$	Numărul de intrări
	FR <sub>j</sub>	0..999	Coeficientul de utilizare (în miimi)
	FT <sub>j</sub>	$2^{31}-1$	Timpul mediu de utilizare de către o tranzacție
Echipamentul multicanal	$R_j$	$2^{31}-1$	Numărul de canale libere
	$S_j$	$2^{31}-1$	Conținutul curent
	SA <sub>j</sub>	$2^{31}-1$	Conținutul mediu
	SC <sub>j</sub>	$2^{31}-1$	Numărul de intrări
	SR <sub>j</sub>	0..999	Coeficientul de utilizare (în miimi)
	SM <sub>j</sub>	$2^{31}-1$	Conținutul maximal
	ST <sub>j</sub>	$2^{31}-1$	Timpul mediu de aflare a unei tranzacții în echipamentul multicanal

Variabila	$BV_j$	0 sau 1	Valoarea variabilei booleene $j$
	$V_j$	$2^{31}-1$	Valoarea variabilei aritmetice $j$
Funcția	$FN_j$	$2^{31}-1$	Valoarea funcției $j$
Șirul de așteptare	$Q_j$	$2^{31}-1$	Lungimea curentă a șirului
	$QA_j$	$2^{31}-1$	Lungimea medie a șirului
	$QM_j$	$2^{31}-1$	Lungimea maximă a șirului
	$QC_j$	$2^{31}-1$	Numărul de intrări
	$QZ_j$	$2^{31}-1$	Numărul de tranzacții cu timpul de reținere nul
	$QT_j$	$2^{31}-1$	Timpul mediu de aflare a tranzacțiilor în șir
	$QX_j$	$2^{31}-1$	Timpul mediu de aflare a tranzacțiilor în șir cu excepția celor cu timp nul
Tabela	$TB_j$	$2^{31}-1$	Valoarea medie a argumentului tabelii
	$TD_j$	$2^{31}-1$	Devierea pătratică medie
	$TC_j$	$2^{31}-1$	Numărul de intrări
Celula	$X_j$	$2^{31}-1$	Valoarea celulei $j$ în format cuvânt
	$XH_j$	$2^{15}-1$	Valoarea celulei $j$ în format semicuvânt
Matricea de celule	$MX(a,b)_j$	$2^{31}-1$	Valoarea elementului din linia $a$ și coloana $b$ în format cuvânt
	$MH(a,b)_j$	$2^{15}-1$	Valoarea elementului din linia $a$ și coloana $b$ în format semicuvânt
Lanțul utilizatorului	$CA_j$	$2^{15}-1$	Conținutul mediu
	$CH_j$	$2^{15}-1$	Numărul curent de elemente în lanț
	$CM_j$	$2^{15}-1$	Numărul maximal de elemente în lanț
	$CC_j$	$2^{31}-1$	Numărul de intrări
	$CT_j$	$2^{31}-1$	Timpul mediu de aflare a unui element în lanț
Atribute de sistem	$C1$	$2^{31}-1$	Timpul relativ al modelului
	$RN1..RN8$	$\{0.0000001 - 0.9999999\}$ <i>sau</i> $\{0 - 999\}$	Generatoare de numere aleatoare în calitate de argument al funcției <i>sau</i> în alte cazuri.

## Cartele de control în GPSS

Cartelele de control în GPSS se folosesc pentru dirijarea cu procesul de modelare.

CARTELELE:

1. **SIMULATE** (*fără operanzi*) – indică simulatorului la necesitatea de a rula modelul. În caz de absență a acestei cartele, după analiza sintactică, simulatorul nu va rula procesul de modelare.

2. **START A,B,C,D** – indică simulatorului că toate datele sunt introduse și este nevoie să se starteze rularea modelului. Câmpul  $A$  indică simulatorului la valoarea care se atribuie contorului de terminări. Procesul de modelare se oprește atunci când acest contor are valoarea mai mică sau egală cu

zero. Contorul de terminări este decrementat cu valoarea care este indicată în câmpul *A* al blocului TERMINATE.

Ex.            TERMINATE        1  
                  START            200

(când 200 de tranzacții vor intra în blocul TERMINATE, procesul modelării va stopa).

Câmpul *B* este folosit pentru extragerea statisticii finale. Dacă în câmpul *B* este indicat NP (Not Print), atunci nu se va extrage statistica finală.

Ex.            GENERATE        200  
                  .....  
                  TERMINATE        1  
                  START            100,NP

Câmpul *C* este folosit pentru primirea statisticii intermediare. În el se indică intervalul de extragere a statisticii. Conținutul câmpului *C* se decrementează sincron cu decrementarea contorului de terminări. Când conținutul câmpului *C* este zero, atunci se extrage statistica intermediară, se inițializează valoarea lui inițială și procesul continuă. Este de dorit ca valoarea câmpului *C* să fie divizibilă la conținutul câmpului *A*. ( $N=i*K$ )

Ex.            START            100,,20

Câmpul *D* indică la necesitatea de a extrage statistica lanțului de evenimente curente, lanțului de evenimente viitoare, lanțului de întreruperi și lanțului utilizatorului. La o astfel de necesitate în câmpul *D* se indică 1, în caz contrar, câmpul *D* este vid.

START            N,[NP],[K],[1]

3. **RESET** – În urma aplicării acestei cartele în model, va fi ștearsă toată statistica. Această cartelă nu acționează asupra celulelor de memorie și matricelor de celule, asupra generatoarelor de numere aleatorii și de asemenea nu elimină tranzacțiile din model. Timpul modelului rămâne neschimbat. Este posibilitate de a folosi cartela în regim selectiv, indicând în câmpurile operanzilor obiectele, atributele cărora nu trebuie să fie schimbate.

Utilizare selectivă:

Ex.            RESET            F1,Q10-Q15

4. **CLEAR** – la aplicarea acestei cartele se anulează întreaga statistică, din model se elimină toate tranzacțiile, timpul modelului devine egal cu zero ( $C1=0$ ), de asemenea sunt vidate toate celulele și matricele de celule. Este posibilă folosirea cartelei în regim selectiv, indicând în câmpul operanzilor obiectele, parametrii cărora trebuie să fie salvați. (Cartela nu acționează asupra generatoarelor de numere aleatorii).

Ex.            CLEAR            X10, X20-X25

5. **JOB** – Această cartelă se introduce între două modele ale unui utilizator. Ea îndeplinește toate funcțiile cartelei CLEAR, dar de asemenea ea mai șterge din memorie descrierea modelului și permite rularea a mai multe modele concomitent. Cartela JOB instalează generatoarele de numere aleatorii în stare inițială.

6. **END** – indică simulatorului la sfârșitul problemelor introduse.

## Metode de adresare în GPSS

Există 2 tipuri de adresări în GPSS: directă și indirectă.

Adresarea directă – adresarea la un obiect concret. Ea este posibilă prin adresarea la numărul sau la numele obiectului.

Ex.            SEIZE            1 ;ocupă dispozitivul nr.1

Numele sau eticheta poate consta din 5 simboluri, dar nu mai puțin de 3, care trebuie să fie litere. La adresarea la un ANS numele lui se separă prin semnul \$.

Ex.            FN\$EXPON

Pentru ANS numerice separatorul nu se indică (FN1).

Adresarea indirectă se folosește atunci, când este nevoie adresarea la un bloc față de alt bloc cu etichetă. De asemenea este posibilă adresarea la blocul curent.

Ex. TRANSFER ,\*+3 ;transferul tranzacției din blocul curent în al 3-ea după el.

În GPSS este posibilă adresarea obiectelor cu ajutorul parametrilor tranzacțiilor. În acest caz în loc de numărul sau numele obiectului se indică ori P<sub>j</sub> ori \*j.

Adresarea la parametrii tranzacțiilor:

$$\begin{cases} *n \\ pn \end{cases}, \text{ pentru adresare directă}$$
$$\{p *n, \text{ pentru adresare indirectă}$$

Pentru primul caz adresarea se face la parametrul cu numărul  $n$  al tranzacției, iar în al doilea – la parametrul cu numărul, indicat în valoarea parametrului  $n$ .

Ex. Valoarea parametrilor tranzacției este: P5=8 și P8=10

SEIZE P5 (sau \*5) ;adr. directă – simulează ocuparea dispozitivului nr.8

SEIZE P\*5 ;adr. indirectă – tranzacția ocupă dispozitivul nr.10

## Variabile în GPSS

În GPSS există 3 tipuri de variabile:

1. Aritmetice
2. Aritmetice cu virgulă flotantă
3. Booleene

1) Variabilele aritmetice prezintă o reuniune aritmetică ale valorilor diferitor ANS. Descrierea variabilei se face prin intermediul unei cartele speciale în formatul următor:

<etichetă> VARIABLE <expresie aritmetică>

Ex. VAR1 VARIABLE Q9+3+FNS\$LINE

Valoarea variabilei VAR1 este calculată în felul următor: lungimea curentă a șirului nr.9+3+valoarea funcției LINE.

Adresarea la funcțiile aritmetice se face cu ajutorul ANS-ului V<sub>j</sub> după nume sau număr. Adresarea la variabila din Ex. precedent se face astfel: V\$VAR1. Este posibilă și adresarea indirectă. (Ex.V\*2)

În GPSS există 5 operații aritmetice:

“+” – adunarea algebrică;

“-” – scăderea algebrică;

“\*” (alte versiuni “#”) – înmulțirea algebrică;

“/” – împărțirea algebrică (partea fracționară se omite);

“@” – împărțirea după modul (restul de la împărțire)[10@7=3];

Până la îndeplinirea oricărei operații, este calculată, în caz de necesitate, valoarea oricărui element (dacă este ANS) și se ia partea lui întregă. La descrierea expresiei este posibilă utilizarea parantezelor. Este posibilă, de asemenea, o altă descriere a variabilei în procesul de modelare (rulare).

2) Variabilele cu virgulă flotantă sunt analogice cu cele aritmetice cu excepția că înainte de îndeplinire a operației, și pe parcursul calculării expresiei, partea fracționară nu se omite. (Partea fracționară se omite numai la valoarea finală a variabilei). Rezultatele intermediare de asemenea nu se schimbă.

Descrierea acestui tip de variabile este analogică cu cea a variabilelor aritmetice simple, cu excepția că în câmpul blocurilor se scrie FVARIABLE. Adresarea se face la fel prin ANS-ul V<sub>j</sub> și numele sau numărul ei trebuie să fie diferit de cel al variabilei aritmetice simple.

Ex. VAR1 VARIABLE 10\*(11/3)

FVAR1 FVARIABLE 10\*(11/3)

de unde reiese că: V\$VAR1=30, iar V\$FVAR1=36

Împărțirea după modul pentru variabilele cu virgulă flotantă este imposibilă.

3) Variabilele booleene prezintă o expresie logică, alcătuită din diferite ANS. Descrierea este analogică cu cea a variabilelor aritmetice, cu excepția că în câmpul blocurilor se scrie BVARIABLE.

La prelucrarea variabilelor booleene simulatorul verifică una sau câteva condiții (rezultatul poate fi 1 sau 0).

Este posibilă utilizarea a 3 operatori pentru acest tip de variabile:

1. Logici
2. Condiționali
3. Booleeni

1) Operatorii logici sunt legați cu stările obiectelor din GPSS. Este posibilă utilizarea următorilor operatori:

<b>DISPOZITIVE</b>	$FU_n(F_n)$	1 – dispozitivul este ocupat 0 – dispozitivul este liber
	$FNU_n$	0 – dispozitivul este ocupat 1 – dispozitivul este liber
	$FI_n$	1 – dispozitivul prelucrează o întrerupere 0 – în caz contrar
<b>ECHIPAMENTE MULTICANAL (EM)</b>	$SF_n$	1 – echipamentul este ocupat (nu sunt canale libere) 0 – în caz contrar (există măcar 1 canal liber)
	$SNF_n$	inversul lui $SF_n$
	$SE_n$	1 – echipamentul este vid 0 – în caz contrar (există măcar 1 canal ocupat)
	$SNE_n$	inversul lui $SE_n$
<b>CHEI LOGICE</b>	$LR_n$	1 – cheia logică este resetată 0 – cheia nu este resetată
	$LS_n$	0 – cheia logică este resetată 1 – cheia nu este resetată

Ex. LOGV BARIABLE SF\$MEM

LOGV=1 atunci când toate canalele EM MEM sunt ocupate și 0 – în caz contrar.

2) Operatorii condiționali sunt folosiți pentru compararea algebrică și sunt scriși în ghilimele unare:

- ‘G’ (Greater) – mai mare (>).
- ‘L’ (Less) – mai mic (<).
- ‘E’ (Equal) – egal (=).
- ‘GE’ (Greater or Equal) – mai mare sau egal ( $\geq$ ).
- ‘LE’ (Less or Equal) – mai mic sau egal ( $\leq$ ).
- ‘NE’ (Not Equal) – diferit ( $\neq$ ).

Ex. LVAR BARIABLE V\$VAR1’G’10

atunci BV\$LVAR=1 atunci când V\$VAR1>10 și BV\$LVAR=0 atunci când V\$VAR1 $\leq$ 10.

3) Pentru operatorii booleeni sunt posibile 2 operații:

“+”(alte versiuni ‘OR’) – operația logică SAU

“\*” (alte versiuni ‘AND’) – operația logică ȘI

Adresarea variabilelor booleene se face cu ajutorul ANS-lui  $BV_j$ , unde numele sau numărul lor poate să corespundă cu cel al celor aritmetice.

Ex. LIBER BARIABLE F1’AND’F2

Variabila LIBER determină starea a 2 dispozitive și BV\$LIBER=0 când măcar un dispozitiv este liber, iar BV\$LIBER=1 când ambele dispozitive sunt ocupate.

## Funcții în GPSS

Ca argument al funcției poate servi orice ANS, inclusiv și o altă funcție. În GPSS este folosită metoda tabelară de definire a funcției și anume cu ajutorul unor perechi de numere ce determină abscisa și valoarea funcției (coordonata unui punct din funcție). Există 5 tipuri de funcții:

1. Funcție numerică continuă (C);

2. Funcție numerică discretă (D);
3. Funcție numerică tabulară (L);
4. Funcție atributivă discretă (E);
5. Funcție atributivă tabulară (M);

Descrierea funcției se face în modul următor:

<etichetă> FUNCTION A,B

Câmpul A conține argumentul funcției, iar B – tipul și numărul de puncte ce descriu funcția dată.

În rândul, ce urmează după cartela FUNCTION, începând cu prima poziție sunt descrise punctele funcției, de către abscisă și valoarea funcției separate prin virgulă. Punctele se separă prin “/”. Valorile absciselor, în descrierea funcției, trebuie să fie aranjate în ordine crescătoare.

Ex.

FUN FUNCTION P3,D5  
1,15/5,10/6,6/9,0/11,50

Argumentul și valoarea funcției pot să primească atât valori întregi cât și fracționare. În momentul adresării la funcție partea întreagă se omite.

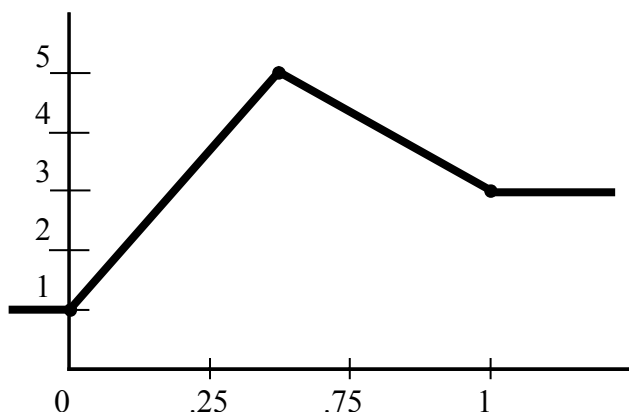
Ex.

FUNC FUNCTION P3,C3  
0,1/0.5,1.75/1,3.89

Adresarea la funcție se îndeplinește prin ANS FN<sub>j</sub> sau FN\$<nume>.

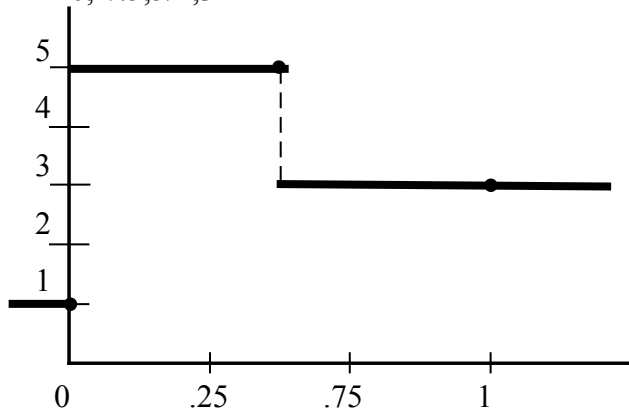
1) Funcția numerică continuă (în câmpul B se indică litera C). Pentru aceste funcții în intervalele ei de definiție se face o interpolare liniară între punctele corespunzătoare. Funcția continuă dă posibilitate de a aproxima orice funcție cunoscută. La nimerirea argumentului în afara intervalului definit, extrapolarea nu se face și funcția primește valoare de la capătul corespunzător al intervalului.

Ex. EXC FUNCTION RN1,C3  
0,1/.5,5/1,3



2) Funcția numerică discretă (în câmpul B se indică litera D). Aceste funcții presupun aceeași valoare a funcției  $FN_j=Y_i$  pe intervalul  $(X_{i-1}, X_i)$ . Interpolarea nu se face, valoarea funcției pe un interval este egală cu valoarea ei la capătul drept al intervalului. Pentru cazul dat nu se admite ca valorile funcției să fie descrise de numere fracționare.

Ex. EXC FUNCTION RN1,C3  
0,1/.5,5/1,3





3) Funcția numerică tabulară (în câmpul B se indică litera L). Dacă valoarea argumentului este 1,2,3,4 ... N (adică numere naturale), atunci așa funcții pot fi descrise ca tabelare (aceasta face economie de timp de calculator). La nimerirea argumentului în afara intervalului [1, N] simulatorul va da un mesaj de eroare [OVERFLOW].

Ex. Descrierea unei funcții pe intervalul [1,10]

FUNL FUNCTION P1,L10  
1,2/2,3/3,2/4,2/5,5/6,5/7,3/8,3/9,3/10,3

3) Funcția atributivă discretă (în câmpul B se indică litera E). Valorile acestei funcții sunt SNA. Funcțiile atributive discrete sunt asemănătoare cu cele numerice discrete.

Ex. FUNE FUNCTION P10,E3

1,Q10/3,C1/10,Q3

4) Funcția atributivă tabulară (în câmpul B se indică litera M). Valorile acestei funcții sunt SNA. Funcțiile atributive tabulare sunt asemănătoare cu cele numerice tabulare.

Ex. FUNM FUNCTION P1,M3

1,X1/2,S34/3,Q15

## Blocuri

### Noțiuni generale.

În urma intrării unei tranzacții într-un bloc pot avea loc următoarele acțiuni:

1. Generarea sau eliminarea tranzacțiilor.

Blocurile GENERATE și SPLIT generează tranzacții (în blocul GENERATE tranzacțiile nu intră).

Blocurile TERMINATE și ASSEMBLE elimină tranzacțiile din model.

2. Schimbarea atributelor obiectelor.

Este posibilă schimbarea atributelor următoarelor obiecte:

a) ale tranzacției ce a intrat în blocul dat

b) ale altor tranzacții

c) ale altor obiecte GPSS

3. Reținerea tranzacției pe decursul unui interval indicat.

### Regulile de bază în descrierea blocurilor.

În câmpurile blocurilor pot fi scrise următoarele înregistrări:

ANS<sub>j</sub> – indică ANS a unui obiect. Ex.: S10

ANS\*n – adresarea indirectă a obiectelor S\*10

C – constantă

Pn – parametru

P\*n – parametru (adresare indirectă)

Cuvinte cheie – BOTH, ALL ș.a.

Str – nume din simboluri (caractere).

Formatul de scriere este următorul:

<etichetă> DEN\_BLOC A,B,C,D,E,F,G

### Blocuri ce operează cu blocuri.

1. Blocul CHANGE – permite de a schimba blocurile în procesul de rulare a modelului. Blocul din câmpul A este schimbat cu blocul indicat în câmpul B.

Formatul:

CHANGE	A	B
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$

Ex.

5 ASSIGN 6,V5

30	..... SAVEVALUE	1,FN4
60	..... GHANGE	5,30

2. Blocul EXECUTE permite executarea a oricăru bloc fără a schimba ordinea de parcurgere a tranzacțiilor. După executarea în blocul indicat tranzacția trece în următorul bloc după blocul EXECUTE.

Formatul:

**EXECUTE      A**

$$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$$

### Generarea și eliminarea tranzacțiilor.

1. Blocul GENERATE este blocul de bază de introducere a tranzacțiilor în model. Nici într-un câmp al acestui bloc nu se permite adresarea indirectă.

Formatul:

**GENERATE    A            B            C            D            E    F    G**

$$\left\{ \begin{array}{l} C \\ ANS_j \end{array} \right\} \left\{ \begin{array}{l} C \\ ANS_j \end{array} \right\} \left\{ \begin{array}{l} C \\ ANS_j \end{array} \right\} \left\{ \begin{array}{l} C \\ ANS_j \end{array} \right\} \{C\} \{C\} [H]$$

Câmpul A este timpul mediu dintre tranzacțiile generate. Câmpul B este modificatorul timpului dintre tranzacții și poate fi de 2 tipuri: interval sau funcție. Modificatorul interval se folosește atunci când timpul de generare a tranzacțiilor este distribuit uniform într-un interval dat. *Observație:* B<=A. În cazul acesta în Câmpul B se indică o constantă sau un oricare ANS(în afară de FN<sub>j</sub>).

Ex.            GENERATE    10,2 ; adică distribuția timpului este în intervalul [8,12].

Modificatorul funcție este folosit când timpul de generare nu este uniform, dar este distribuit conform unei legi. În acest caz pentru determinarea timpului concret a valorii timpului, conținutul câmpului A se înmulțește la conținutul câmpului B. Funcția trebuie să fie descrisă anterior (prin cartela FUNCTION). De regulă funcția în acest caz are ca argument un generator de numere aleatoare. În momentul adresării la funcție, simulatorul calculează valoarea ei și o înmulțește cu conținutul din câmpul A.

Ex.            GENERATE    10,FN\$MYFUN

Câmpul C este întârzierea inițială. Acesta indică intervalul de timp după care vor începe a se genera tranzacții. Câmpul D va indica numărul de tranzacții care vor fi generate în model. Dacă câmpul D este vid, atunci numărul de tranzacții generate este infinit. Câmpul E arată care este prioritatea tranzacțiilor generate. C=0..127. Implicit prioritatea tranzacțiilor este nulă. Câmpul F indică numărul de parametri ce vor caracteriza tranzacția. C=0..100. Implicit numărul de parametri este egal cu 12. Și câmpul G arată formatul parametrilor. Implicit este F (cuvânt de 32 biți), iar după dorință putem indica H (semicuvânt de 16 biți).

Ex.            GENERATE    X3,FN\$MYFUN,8,1000

2. Blocul TERMINATE este destinat pentru eliminarea tranzacțiilor din model. Câmpul A indică numărul de unități care se vor decrementa din contorul de terminări la intrarea fiecărei tranzacții în acest bloc.

Formatul:

**TERMINATE    A**

$$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$$

În model pot fi un număr nelimitat de blocuri terminate.

### Dirijarea timpului de modelare.

1. Blocul ADVANCE simulează reținerea tranzacției în model. În câmpul A este indicat timpul mediu de reținere a tranzacției și câmpul B este modificatorul acestui timp. Sensul modificatorului este ca și în cazul blocului GENERATE.

Formatul:

$$\begin{array}{ccc} \text{ADVANCE} & \text{A} & \text{B} \\ \left. \begin{array}{l} C \\ \text{ANS}_j \\ \text{ANS} * n \\ * n \end{array} \right\} & & \left. \begin{array}{l} C \\ \text{ANS}_j \\ \text{ANS} * n \\ * n \end{array} \right\} \text{ sau } \left. \begin{array}{l} \text{FN}_j \\ \text{FN} * n \end{array} \right\} \end{array}$$

*Observație:* Constantele nu pot depăși valoarea  $C \leq 999999$ . Acest bloc poate să prelucreze în paralel mai multe tranzacții.

### Dispozitive și blocurile lor corespunzătoare.

Dispozitivele sunt folosite pentru simularea funcționării echipamentelor cu un canal de deservire. Corespunzător în dispozitiv se poate afla doar o singură tranzacție.

1. Blocul SEIZE este folosit pentru simularea ocupării dispozitivului de o tranzacție. În caz dacă dispozitivul este ocupat înaintea acestui bloc se creează un șir de așteptare. În cazul sosirii unei tranzacții cu prioritate mai mare, tranzacția care se află în acest bloc va fi deservită până la capăt, iar după eliberarea dispozitivului din șirul de așteptare va fi deservită acea tranzacție care are prioritate mai mare. Restul vor fi deservite în dependență de timpul lor de sosire.

Formatul:

$$\begin{array}{ccc} \text{SEIZE} & \text{A} & \\ \left. \begin{array}{l} C \\ \text{ANS}_j \\ \text{ANS} * n \\ * n \\ \text{Str} \end{array} \right\} & & \end{array}$$

2. Blocul RELEASE se utilizează pentru a simula eliberarea dispozitivului de către tranzacția care l-a ocupat. Eliberare se înfăptuiește în momentul intrării tranzacției în blocul dat. Acest bloc nu reține niciodată tranzacția la intrare.

Formatul:

$$\begin{array}{ccc} \text{RELEASE} & \text{A} & \\ \left. \begin{array}{l} C \\ \text{ANS}_j \\ \text{ANS} * n \\ * n \\ \text{Str} \end{array} \right\} & & \end{array}$$

Dispozitivul trebuie să fie eliberat de acea tranzacție care l-a ocupat, în caz contrar simulatorul va da eroare.

3. Blocul PREEMPT îndeplinește o funcție asemănătoare cu a blocului SEIZE cu excepția că tranzacția care intră în acest bloc, poate, în caz de necesitate, să întrerupă deservirea tranzacției care deja a ocupat dispozitivul dat, după prioritate și în continuare să fie deservită ea însăși de către dispozitiv. Tranzacția care a întrerupt dispozitivul dat nu poate să întrerupă, în același timp, alt dispozitiv. Dacă dispozitivul nu este ocupat sau întrerupt atunci PREEMPT funcționează la fel ca și blocul SEIZE. Dacă dispozitivul este întrerupt de o tranzacție cu aceeași prioritate, atunci înaintea lui se creează un șir de așteptare pentru întreruperi de aceeași clasă.

Dacă se dorește ca blocul PREEMPT să funcționeze la fel ca blocul SEIZE atunci el are următorul format.

$$\begin{array}{ccc} \text{PREEMPT} & \text{A} & \\ \left. \begin{array}{l} C \\ \text{ANS}_j \\ \text{ANS} * n \\ * n \\ \text{Str} \end{array} \right\} & & \end{array}$$

În alt caz,

**PREEMPT**      **A**      **B**      **C**      **D**      **E**

$$\left. \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \\ Str \end{array} \right\} PR \left. \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \\ Str \end{array} \right\} \left. \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\} [RE]$$

Una din cele mai importante funcții ale blocului PREEMPT este posibilitatea de a întrerupe deservirea tranzacției de o tranzacție cu prioritate mai înaltă. Pentru folosirea acestui regim (prioritar) se indică în câmpul B ANS-ul PR. În acest regim tranzacția cu prioritate  $n$  poate fi întreruptă de o tranzacție cu prioritate mai mare decât  $n$ . În câmpul C se indică numărul (numele) blocului unde va fi transmisă tranzacția întrerupă, care în continuare va pretinde la acest dispozitiv. Dacă în câmpul C nu este indicat nimic, atunci tranzacția este inclusă în lanțul de întreruperi. În câmpul D poate fi scris un ANS (constantă) valoarea căruia determină numărul parametrului a tranzacției întrerupte în care se va scrie timpul rămas pentru deservire în acest dispozitiv. În câmpul E se poate indica RE ce înseamnă că tranzacția întreruptă în continuare nu va pretinde la dispozitivul dat. Este necesar de semnalizat că dacă câmpul B este vid atunci celelalte ce urmează după el sunt ignorate.

4. Blocul RETURN este folosit pentru simularea anulării întreruperii de pe dispozitivul întrerupt. Este analogul blocului RELEASE.

Formatul:

**RETURN**      **A**

$$\left. \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \\ Str \end{array} \right\}$$

### Echipamente multicanal (EM). Noțiuni generale

Obiectele din tipul EM de obicei sunt folosite pentru simularea funcționării dispozitivelor cu deservire paralelă (de ex. memoria PC). Dispozitivele de acest tip sunt caracterizate de mărime și pot paralel deservi un număr determinat de tranzacții fiecare din ele ocupând un număr oarecare de canale. EM blochează intrarea tranzacțiilor, în caz de absența a canalelor libere sau neajuns de canale.

Statistica finală conține următoarele caracteristici pentru EM:

- numărul mediu de canale ocupate;
- coeficientul de utilizare a EM;
- numărul de intrări în EM;
- timpul mediu de reținere a unei tranzacții în EM;
- conținutul EM în momentul de sfârșire a simulării;
- conținutul maximal;

(EM dețin de 7 ANS și 4 operatori logici)

Mărimea oricărui EM trebuie să fie descris înaintea procesului de simulare. La absența descrierii se va lua mărimea lui maxim posibilă ( $2^{31}-1$  de canale). Descrierea se înfăptuiește prin intermediul cartei STORAGE. Sunt posibile 2 formate ale acestei cartei:

a) <etichetă> **STORAGE**    N

unde eticheta este numele EM, iar N este un număr ce indică numărul de canale.

Ex. MEM STORAGE      256

b) în acest caz câmpul etichetei este vid, câmpul blocurilor conține cartela STORAGE, iar câmpul operanzilor conține descrierea EM. În acest caz, printr-o cartelă se pot descrie mai multe EM despărțite prin “/”.

Ex. STORAGE      S\$MEM,256

Pentru descrierea a mai multe EM cu aceiași mărime ele se despart prin “-”.

Ex. STORAGE      S\$MEM,256/S1-S5,100

Blocurile ce funcționează cu EM sunt:

1. Blocul ENTER simulează ocuparea a unui sau a mai multe canale la intrarea în acest bloc a tranzacției. Deține de proprietatea de a se bloca la neajunsul de canale libere. Câmpul A indică numele sau numărului EM, iar câmpul B – numărul de canale pe care le ocupă tranzacția intrând în acest bloc (implicit se ocupă un canal).

Formatul:

<b>ENTER</b>	<b>A</b>	<b>B</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$

Ex.            ENTER            5 ;a ocupa un canal din EM nr.5  
                   ENTER            \*6,10 ;a ocupa 10 canale din EM indicat în P6 (parametru 6)  
                   ENTER            1,R1 ;a umple EM (R1 – numărul de canale libere)

2. Blocul LEAVE este folosit pentru eliberarea unui număr dat de canale. Câmpurile A și B sunt analogice blocului ENTER. Niciodată nu refuză tranzacției la intrare.

Formatul:

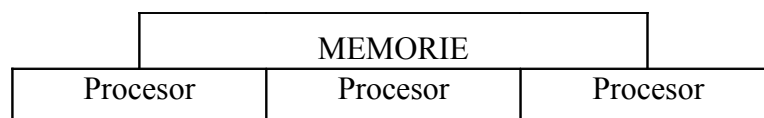
<b>LEAVE</b>	<b>A</b>	<b>B</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$

Implicit se eliberează un canal. Dacă conținutul câmpului B este mai mare decât conținutul curent al EM atunci va apărea eroare. Conținutul curent  $S_j$  al EM se micșorează cu mărimea din câmpul B, la intrarea tranzacției în blocul LEAVE, iar  $R_j$  se mărește cu această mărime.

$S_j' = S_j - [B]$ ,     $R_j' = R_j + [B]$

Ex.            LEAVE            5 ;a elibera un canal din EM nr.5  
                   LEAVE            \*6,10 ;a elibera 10 canale din EM indicat în P6 (parametru 6)  
                   LEAVE            1,S1 ;a elibera deplin EM (S1 – numărul de canale ocupate)

Cel mai simplu exemplu de sistem multiprocesor cu memorie comună:



```

SIMULATE
RND      FUNCTION      RN2,D3
0.333,1/.667,2/1,3
MEM      STORAGE      512
          GENERATE      100,10
          ASSIGN        1,FN$RND
          ENTER         MEM
          SEIZE         P1
          ADVANCE      230,30
          RELEASE      P1
          LEAVE         MEM
TERMINATE 1
START 1000
END
  
```

## Blocurile categoriei de statistică.

### Șiruri de așteptare și blocurile lor corespunzătoare. Noțiuni generale.

Obiectele de tipul șir de așteptare sunt folosite pentru culegerea statisticii despre cozile care apar în sistemul modelat în rezultatul diferitor condiții de blocare. În statistica extrasă de către simulator putem distinge:

- lungimea medie a cozii;
- numărul de intrări în coadă;
- numărul de intrări fără rețineri;
- timpul mediu de reținere a unei tranzacții în coadă;
- timpul mediu de reținere a unei tranzacții în coadă luând în considerație cele fără rețineri; (fără ele)
- lungimea curentă și maximală a cozii;

1. Blocul QUEUE este folosit pentru înregistrarea intrării tranzacției în coadă. În câmpul A se indică numele (numărul) cozii, iar în câmpul B – numărul de locuri (unități) pe care le ocupă tranzacția în șirul de așteptare. (Implicit șirul se incrementează cu o unitate).

Formatul:

QUEUE	A	B
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$

Blocul QUEUE nu refuză niciodată tranzacției la intrare. De obicei după el stau blocuri cu caracteristici de blocare (ca SEIZE, PREEMPT, ENTER ș.a.). Tranzacția poate simultan să se afle în câteva șiruri.

Ex.            QUEUE            3  
                  QUEUE            \*1,P3

2. Blocul DEPART decrementează lungimea șirului de așteptare indicat în câmpul A cu numărul de unități indicat în câmpul B.

Formatul:

DEPART	A	B
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$

Conținutul câmpului B trebuie totdeauna să fie mai mic decât lungimea curentă a cozii.

Ex.            DEPART            QUE

### Tabele. Noțiuni generale.

Tabelele sunt folosite pentru primirea funcțiilor de repartiție a argumentului dat, care poate orișicare ANS. Statistica de bază reprezentată de tabele este următoarea:

- suma valorilor argumentului, introduse în tabel după ultima activare a cartelei CLEAR sau RESET;
- numărul de intrări în tabelă;
- valoarea medie a argumentului, care se calculează ca sumă a tuturor valorilor, împărțită la numărul de intrări;
- devierea pătratică medie a argumentului;
- valoarea medie a argumentului în ultimul interval.

Intervalul tablei în statistică constă din următoarele:

1. limita de sus a intervalului;
2. numărul de nimeriri în interval;
3. procentajul e nimerire în interval;

4. numărul sumar de valori mai mici sau egale cu limita de sus a intervalului;
5. numărul sumar de valori mai mari decât limita de sus a intervalului.

Cartela TABLE este folosită pentru descrierea unei tabele. Această cartelă are un șir de câmpuri:

Formatul:

```
<etichetă> TABLE      A      B C D [E]
                        {C
                        ANSj
                        ANS* n
                        * n }
```

unde câmpul A este argumentul tabelii, B – limita de jos a intervalului, C – pasul intervalului, D – numărul de intervale. [Pentru regimul RT se folosește de asemenea câmpul E care redă intervalul de tabulare].

Tabelele pot funcționa în regim de diferențiere care este determinat de semnul minus după argument. În acest caz în tabelă nu se include valoarea absolută a argumentului, dar diferența dintre valoarea curentă și precedentă. La prima nimerire în interval în tabelă nu se include nimic, se memorează doar valoarea argumentului.

Există 3 regimuri speciale de funcționare a acestei cartele: IA, RT, QTABLE. Pentru folosirea lor, în câmpul argumentului se scrie mnemonica corespunzătoare regimului.

Ex. XTIME TABLE M1,0,10,100

Blocul TABULATE. Valoarea argumentului tabelii se introduce în ea de fiecare dată când tranzacția intră în blocul TABULATE. Numele tabelii se indică în câmpul A. În câmpul B se indică numărul de unități care va incrementa contorul de nimerire în intervalul argumentului (implicit este 1).

Formatul:

```
TABULATE  A      B
          {C
          ANSj
          ANS* n
          * n }
```

Ex. TABULATE XTIME

### Regimuri speciale de funcționare a tabelilor

Regimul IA (Inter Arival) este folosit pentru evaluarea distribuției intervalelor de timp dintre tranzații într-un punct anumit al modelului. În acest caz, la nimerirea tranzației în blocul TABULATE se fixează timpul absolut al modelului și din el se scade timpul de sosire a tranzației precedente. Rezultatul primit se include în tabelă.

Ex. DISP TABLE IA,0,100,10

Regimul RT (Rate) este folosit pentru evaluarea distribuirii intensității (ratei) sosirii tranzațiilor într-un punct anumit al modelului. Pentru folosirea lui, în câmpul argumentului se indică mnemonica RT, iar blocul TABULATE în punctul din model care ne interesează. Acest regim necesită includerea câmpului E, care indică intervalul de timp în care se va măsura rata.

Regimul QTABLE permite de a evalua regula de distribuire a timpului de aflare a tranzațiilor în șirul de așteptare. De obicei, fără acest regim, aceasta se face în felul următor:

```
TAB TABLE MP2,0,5,40
```

```
.....
MARK      2
QUEUE     8
SEIZE     3
DEPART    8
TABULATE  TAB
```

iar în folosind regimul QTABLE aceasta ar fi astfel:

```
TAB QTABLE 8,0,5,40
```

.....

QUEUE	8
SEIZE	3
DEPART	8
TABULATE	TAB

### Blocuri, care schimbă direcția de mișcare a tranzacțiilor

1. Blocul TRANSFER este mijlocul de bază pentru schimbarea direcției e mișcare a tranzacției. Este posibilă funcționarea lui în 9 regimuri, care se indică în câmpul A:

Regimul	Câmpul A
Necondiționat	Vid
Statistic	.număr din 3 cifre
BOTH	BOTH
ALL	ALL
PICK	PICK
Funcție	FN
Parametric	P
Subprogram	SBR
SIM	SIM

Câmpul A indică regimul acestui bloc, iar B și C indică la blocurile posibile de transfer a tranzacțiilor.

1.1. Regimul necondiționat. Tranzacția care intră în acest bloc va fi transmis blocului indicat în câmpul B (câmpul C nu se utilizează în acest regim).

Ex. TRANSFER ,NEXT

1.2. Regimul statistic. Valoarea argumentului, care urmează după punct (până la 3 cifre), indică ce parte din tranzacții vor fi transferate în blocul indicat în câmpul C. Cealaltă parte din ele vor fi transferate în blocul din câmpul B. Dacă câmpul B este vid, atunci partea aceasta (care-i corespunde câmpului B) se transferă următorului bloc ce se află după blocul TRANSFER. În acest regim este posibilă adresarea indirectă. Înregistrarea de tipul .\*n indică că această probabilitate se află în parametrul n a tranzacției.

În GPSS proporția (partea de tranzacții) poate fi atribuită cu ajutorul oricărui ANS. La calcularea ANS-lui, pot apărea 3 cazuri. Fie X valoarea ANS-lui. Atunci avem:

a)  $X \leq 0$

b)  $0 < X < 1000$

c)  $X \geq 1000$

În cazul a) tranzacția care intră în acest bloc este transmisă blocului indicat în câmpul B.

În cazul b) X determină partea ce se transmite.

În cazul c) tranzacția va trece în blocul indicat în C.

Ex. TRANSFER .333,11,25 ;33.3% din tranzacții trec în blocul 25, iar 66.7% în 11.  
TRANSFER .X10,BLOC1,BLOC3

1.3. Regimul BOTH. În acest regim fiecare tranzacție care intră în blocul TRANSFER va încerca să treacă în blocul indicat în câmpul B. Dacă încercarea a avut eșec (intrarea în bloc este blocată), se va produce o încercare de a intra în blocul indicat în câmpul C. În caz de refuz, tranzacția va rămâne în blocul TRANSFER și va încerca să iasă din el (în aceeași modalitate) după schimbarea următoare a timpului modelului.

Ex. TRANSFER BOTH,BLOC1,TERM

1.4. Regimul ALL. Orice tranzacție ce nimerește în blocul TRANSFER, mai întâi încearcă să treacă în blocul (fie că are numărul N) indicat în câmpul B. În caz de refuz, tranzacția va încerca consecutiv să treacă în blocurile  $N+M$ ,  $N+2*M$ , până la L, unde M este pasul ce se indică în câmpul D, iar L este numărul blocului ce se indică în câmpul C. În acest caz  $(L-N)$  este divizibil la M;  $L > N$ ;  $L = i*M + N$ . În caz de refuz în toate direcțiile, tranzacția rămâne în lanțul evenimentelor curente până la apariția condițiilor de transfer. În acest regim nu se permite utilizarea adresării indirecte și a atributelor ANS în câmpurile B,C,D.

Ex. TRANSFER ALL,60,120,10  
TRANSFER ALL,NEXT1,NEXT2,5



1.5. Regimul PICK. Din consecutivitatea blocurilor cu numerele N, N+1, N+2..., N+i, M, în mod aleator se alege blocul pentru transferarea tranzacției. N este numărul blocului din câmpul B, M este numărul din câmpul C. Probabilitatea de a alege unul din aceste blocuri este aceeași.  $p=1/(M-N+1)$   
 Observație: Tranzacția va încerca să treacă numai în blocul selectat (ce s-a calculat). În urma refuzului în acel bloc, tranzacția va rămâne în blocul TRANSFER până la deblocarea acestei direcții.

Ex. TRANSFER PICK,15,19

1.6. Regimul funcție. În acest regim simulatorul calculează valoarea funcției, numele căreia este indicat în câmpul B și aruncă partea ei fracționară. Pentru determinarea numărului blocului unde va fi transferată tranzacția, valoarea obținută a funcției se adună la valoarea indicată în câmpul C. În caz de blocare, tranzacția rămâne în blocul TRANSFER, așteptând condițiile necesare.

Ex. TRANSFER FN,15,\*5 ;Numărul blocului va fi egal cu FN15+P5.

1.7. Regimul parametric. În câmpul A se scrie "P". Valoarea argumentului din câmpul B a blocului TRANSFER în acest caz se interpretează ca numărul parametrului ce intră în acest bloc. Pentru determinarea numărului blocului unde va fi transferată tranzacția se adună valoarea acestui parametru cu argumentul din câmpul C.

Ex. TRANSFER P,5, FN15 ;Numărul blocului va fi egal cu P5+FN15.

1.8. Regimul subprogram. În câmpul A se scrie "SBR". Tranzacția care intră în blocul TRANSFER în acest regim încearcă să treacă în blocul indicat în câmpul B. Valoarea câmpului C se interpretează ca număr al parametrului, în care automat se va scrie numărul de ordine a blocului TRANSFER. Acest regim, de obicei, se folosește ca transfer la un subprogram care se începe din poziția indicată în câmpul B.

Ex. TRANSFER SBR,NEXT,10

.....  
 NEXT -----  
 <textul subprogramului>  
 TRANSFER P,10,1

1.9. Regimul SIM. În acest regim mișcarea tranzacției va fi întreruptă din cauza conectării indicatorului de reținere. Acest indicator se va afla în stare conectată până la îndeplinirea anumitor condiții.

2. Blocul LOOP este folosit pentru organizarea ciclurilor (buclelor) în GPSS. În calitate de contor al numărului de cicluri poate fi folosit orice parametru al tranzacției. Acest bloc niciodată nu dă refuz tranzacției care intră în el. Simulatorul calculează valoarea parametrului tranzacției, indicat în câmpul A, la intrarea ei în blocul LOOP. Această valoare se decrementează cu o unitate și este atribuită din nou parametrului dat. în GPSS. În calitate de contor al numărului de cicluri poate fi folosit orice parametru al tranzacției. Acest bloc niciodată nu dă refuz tranzacției care intră în el. Simulatorul calculează valoarea parametrului tranzacției, indicat în câmpul A, la intrarea ei în blocul LOOP. Această valoare se decrementează cu o unitate și este atribuită din nou parametrului dat. Astfel tranzacția este transmisă blocului indicat în câmpul B, dar numai în cazul când această valoare este diferită de zero, în caz contrar tranzacția va trece în blocul următor.

Formatul:

<b>LOOP</b>	<b>A</b>	<b>B</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \\ Str \end{array} \right\}$

Ex. LOOP 5,ADR1

3. Blocul TEST indică numărul blocului următor pentru transferul tranzacției în cazul îndeplinirii sau neîndeplinirii a anumitor condiții. Blocul TEST analizează relația algebrică dintre două ANS care sunt indicate în câmpurile A și B. Relația se redă prin mnemonica: G,L,E,NE,GE,LE care se includ implicit după blocul TEST. Blocul lucrează în două regimuri: cu intrare condiționată și necondiționată.

În regimul cu intrare necondiționată (este prezent câmpul C), tranzacția nu se reține niciodată la intrare. Dacă condiția se îndeplinește, tranzacția trece în următorul bloc după TEST, iar în caz contrar în blocul indicat în câmpul C.

În regimul cu intrare condiționată (câmpul C este vid), tranzacția nu poate intra în blocul TEST până când nu se va îndeplini condiția dată. Dacă condiția se îndeplinește, tranzacția va trece în blocul ce urmează blocului TEST.

Formatul:

<b>TEST cond</b>	<b>A</b>	<b>B</b>	<b>C</b>
$\left\{ \begin{array}{l} E \\ NE \\ G \\ GE \\ L \\ LE \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$

Ex.            TEST L                    C1,1000

4. Blocul GATE ca și blocul TEST permite schimbarea direcției de mișcare a tranzacției în dependență de 12 atribute logice (relații). Această relație se scrie implicit după GATE. Numărul obiectului verificat este indicat în câmpul A. Acest bloc ca și precedentul are două regimuri de funcționare la intrare (adresa alternativă de transfer se indică în câmpul B).

Mnemonica atributelor logice:

pentru dispozitive și echipamente multicanal(EM):

- |  |  |
|--|--|
| <ul style="list-style-type: none"> <li>• U – dispozitiv ocupat;</li> <li>• NU – dispozitiv liber;</li> <li>• I – dispozitiv întrerupt;</li> <li>• NI – dispozitiv neîntrerupt;</li> <li>•</li> </ul> | <ul style="list-style-type: none"> <li>• SE – EM vid (<math>S_j=0</math>);</li> <li>• SNE – EM nu este vid;</li> <li>• SF – EM plin (<math>R_j=0</math>);</li> <li>• SNF – EM nu este plin;</li> </ul> |
|--|--|

pentru chei logice:

- LS – cheia este setată;
- LR – cheia este resetată;

pentru tranzacții:

- M – în blocul indicat în câmpul A, se află o tranzacție în stare de sincronizare, care aparține aceleiași familii ca și tranzacția care intră (sau încearcă să intre) în blocul GATE.
- NM – în blocul indicat în câmpul B, nu sunt tranzacții în stare de sincronizare și din aceeași familie ca și tranzacția ce intră în blocul GATE.

Dacă se verifică aceste trei tipuri de obiecte se preferă utilizarea blocului GATE și nu TEST pentru că este mai rapid în funcționare.

Formatul:

<b>GATE cond</b>	<b>A</b>	<b>B</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$

Ex.            GATE NU                    5

### Blocuri, care schimbă parametrii tranzacțiilor

1. Blocul ASSIGN se folosește pentru atribuirea valorilor numerice parametrilor tranzacțiilor. Numărul parametrului se indică în câmpul A, în câmpul B se scrie valoarea nouă care i se atribuie parametrului, iar în câmpul C poate fi indicat modificatorul funcție și în acest caz valoarea nouă a

parametrului se determină prin înmulțirea conținutului câmpului B la funcția corespunzătoare indicată în câmpul C. Blocul ASSIGN poate să funcționeze în trei regimuri: increment, decrement și înlocuire. Pentru funcționarea în regim increment în câmpul A implicit după argument se scrie semnul "+". Valoarea nouă a parametrului se va calcula prin adăugarea la valoarea precedentă a conținutului din câmpul B. Regimul decrement se activează prin scrierea semnului "-" după argumentul din câmpul A. Valoarea nouă este determinată prin scăderea valorii curente a parametrului, cu conținutul din câmpul B. Pentru regimul de înlocuire în câmpul A după argument nu este nimic. Valoarea nouă care este indicată în câmpul B va înlocui valoarea curentă a parametrului tranzacției.

Formatul:

<b>ASSIGN</b>	<b>A</b>	<b>B</b>	<b>C</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$

Ex.      **ASSIGN**            5,10  
           **ASSIGN**            \*5+,X15

2. Blocul INDEX este folosit pentru schimbarea primului parametru al tranzacției care intră în acest bloc. Valoarea câmpului B se adună la valoarea parametrului indicat în câmpul A și rezultatul obținut este atribuit primului parametru. Dacă nu este indicat incrementul (câmpul B) el se socotește egal cu zero.

Formatul:

<b>INDEX</b>	<b>A</b>	<b>B</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$

Ex.      **INDEX**            1,1 ;P1=P1+1  
           **INDEX**            3,P5 ;P1=P3+P5

3. Blocul MARK. Orice tranzacție are două ANS care sunt legate cu intervalele de timp de parcurgere prin model. Unul este M1 – timpul tranzacției ( $M1=C1-T_{gen}$ ) și altul este MP1 – timpul parametric a tranzacției ( $MPn=C1-Pn$ ) și  $M1 < MP1$ .

La intrarea tranzacției în blocul MARK parametrului, indicat în câmpul A, i se atribuie valoarea timpului modelului. Dacă câmpul A este vid această valoare se atribuie unui parametru intern pentru marcarea timpului.

Formatul:

<b>MARK</b>	<b>A</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$

Ex.      **MARK**            10  
           **MARK**            \*5

4. Bloc COUNT permite utilizatorului de a număra obiectele care satisfac o anumită condiție. Câmpul A indică numărul parametrului tranzacției în care se organizează un numărător de obiecte ce satisfac condiția dată. Câmpurile B și C determină intervalul de scanare a obiectelor (B – limita de jos, C – limita de sus). Câmpul D se folosește cu operatorii condiționali: E,NE,G,GE,L,LE care se scriu nemijlocit după blocul COUNT. În acest caz valoarea ANS din câmpul D se compară cu valoarea ANS din câmpul E. Câmpul E indică un ANS al obiectelor cercetate care în continuare se va compara cu argumentul din câmpul D. În câmpul E se scrie numai mnemonica ANS, deoarece intervalul de numărare este redat de B și C. La utilizarea operatorilor logici (ca și pentru blocul GATE în afară de M și NM), câmpurile D, E nu se folosesc.

Formatul:

<b>COUNT cond</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j^{19} \\ ANS * n \\ * n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS * n \\ * n \end{array} \right\}$

Ex.       COUNT LE     1,1,5,X10,FC  
           COUNT SF     5,10,30  
           COUNT NU    10,X1,X2

5. Blocul SELECT funcționează analogic blocului COUNT, cu excepția că în loc de numărare a obiectelor care satisfac o anumită condiție, acest bloc selectează primul din ele, scriind numărul lui în parametrul indicat în câmpul A. Câmpurile A...E sunt analogice câmpurilor corespunzătoare ale blocului COUNT. Câmpul F permite ieșirea alternativă în cazul când în diapazonul de scanare nu sunt obiecte, care satisfac condiția dată. Dacă câmpul F este vid, atunci tranzacția totdeauna trece în blocul următor. Spre deosebire de blocul COUNT, blocul SELECT poate funcționa în regimurile MIN și MAX. În acest caz în blocul SELECT sunt selectate acele obiecte (sau un obiect), atributele cărora sunt minimale (maximale). De asemenea, în acest caz câmpurile D și F sunt vide.

Formatul:

<b>SELECT cond</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$

Ex.       SELECT MAX   1,5,10,,FR  
           SELECT GE    10,10,20,X5,Q  
           SELECT NU    1,1,10,,NEXT

### Blocurile PRIORITY și BUFFER

Blocul PRIORITY este destinat pentru atribuirea priorității necesare tranzacției. Prioritatea influențează la starea (locul) tranzacției în lanțul evenimentelor curente și corespunzător la ordinea lui de prelucrare și de asemenea la locul lui în șirurile de așteptare. Prioritatea inițială a tranzacției se instalează în blocul GENERATE în câmpul E. În blocul PRIORITY prioritatea necesară se scrie în câmpul A. Dacă câmpul B este vid, simulatorul va schimba prioritatea tranzacției și-l va mișca mai departe până la oprirea lui. Dacă în câmpul B este mnemonica BUFFER, atunci îndată după schimbarea priorității tranzacției, simulatorul va schimba lanțul evenimentelor curente în corespundere cu noile priorități și va începe prelucrarea lui din nou.

Formatul:

<b>PRIORITY</b>	<b>A</b>	<b>B</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	[BUFFER]

Blocul BUFFER este folosit pentru oprirea imediată de prelucrare a lanțului evenimentelor curente și a inițializării unei noi prelucrări.

Formatul:

**BUFFER**

### Familii de tranzacții și blocurile corespunzătoare

1. Blocul SPLIT este folosit pentru crearea unui număr anumit de copii a tranzacției care intră în acest bloc. Spre deosebire de tranzacțiile introduse prin blocul GENERATE, copiile create prin blocul SPLIT, formează o familie cu tranzacția-original. Numărul necesar de copii se indică în câmpul A. Dacă acest câmp este 0, atunci blocul SPLIT nu face nimic, numai transmite tranzacția mai departe. Copiile sunt create în momentul intrării tranzacției-părinte în blocul SPLIT și apoi tranzacția-părinte încearcă să treacă în blocul următor, indicat în câmpul B, cu atât mai mult valoarea argumentului câmpului B poate fi calculată pentru fiecare copie în parte. În câmpul C poate fi indicat numărul parametrului, utilizat pentru atribuirea numărului de ordine fiecărei copii în ordine crescătoare. De exemplu, dacă valoare acestui parametru pentru tranzacția-părinte este 0 atunci numerotare copiilor va fi de la 1,2 ...A. ( $Pn_i = Pn + i$ ), unde  $Pn$  – valoarea parametrului tranzacției-părinte,  $i$  – numărul copiei. Câmpul D se folosește pentru atribuirea numărului de parametri pentru fiecare copie. Dacă acest câmp

este vid atunci numărul de parametri a copiilor corespunde cu numărul de parametri pentru tranzacția-părinte și de asemenea se păstrează și valorile lor.

Cu ajutorul atributelor interne se organizează o listă a membrilor familiei. Simulatorul urmărește componența familiei, eliminând din ea tranzacțiile ce au fost eliminate din model. În același timp există o mulțime de familii deoarece fiecare tranzacție introdusă în model de blocul GENERATE constituie în sine o familie.

Formatul:

<b>SPLIT</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$[C]$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$

Ex.        SPLIT            3,NEXT,7  
              SPLIT            4,FN\$ADR,5,10

2. Blocul ASSEMBLE assemblează (cuplează) un număr dat de tranzacții dintr-o familie. Tranzacțiile nu sunt reținute niciodată la intrarea în acest bloc. După sosirea numărului necesar de tranzacții dintr-o familie (numărul este indicat în câmpul A), din acest bloc va ieși numai prima tranzacție care a intrat în acest bloc, iar celelalte vor fi eliminate din model.

Particularități:

- Pentru fiecare familie într-un bloc ASSEMBLE se face o singură operație de cuplare.
- Într-un bloc pot fi cuplate simultan (paralel) tranzacții din diferite familii.
- Pentru orice familie cuplarea poate fi îndeplinită paralel în diferite blocuri ASSEMBLE.
- După sfârșirea operației de cuplare pentru o familie dată, poate începe o nouă cuplare din această familie.

Formatul:

<b>ASSEMBLE</b>	<b>A</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$

Ex.        ASSEMBLE        12  
              ASSEMBLE        P2

3. Blocul GATHER este analogic blocului ASSEMBLE în sensul prelucrării familiilor de tranzacții și se deosebește de el prin faptul că acest bloc nu elimină tranzacțiile din model, dar permite mișcarea lor mai departe la apariția ultimei tranzacții necesare pentru cuplare.

Formatul:

<b>GATHER</b>	<b>A</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$

4. Blocul MATCH este folosit pentru sincronizarea mișcării a două tranzacții din aceeași familie fără eliminarea lor din model. Blocul MATCH nu cuplează tranzacțiile, ci permite mișcarea lor în propria direcție. Sincronizarea se efectuează prin alegerea perechilor de tranzacții din aceeași familie și reținerea lor până când ambele vor atinge anumite puncte din model. Tranzacțiile niciodată nu sunt reținute la intrarea în acest bloc și trec în următorul bloc în cazul îndeplinirii condiției de sincronizare. În câmpul A se indică numele (numărul) celuiilalt bloc, care se numește conjugat (blocurile indică unul altuia). Dacă în așa bloc nimerește o tranzacție, care este singurul membru al familiei, atunci se va genera eroare. Este posibilă utilizarea blocului MATCH conjugat cu el însuși. În acest caz funcționarea lui este echivalentă cu blocul GATHER cu valoarea din câmpul A egală cu 2.

Ex. META MATCH            META

La intrarea tranzacției în blocul MATCH, simulatorul verifică blocul conjugat lui, apoi verifică familia acestei tranzacții și în caz de prezență în blocul conjugat a unei tranzacții din aceeași familie, va permite mișcarea lor mai departe.

Formatul:

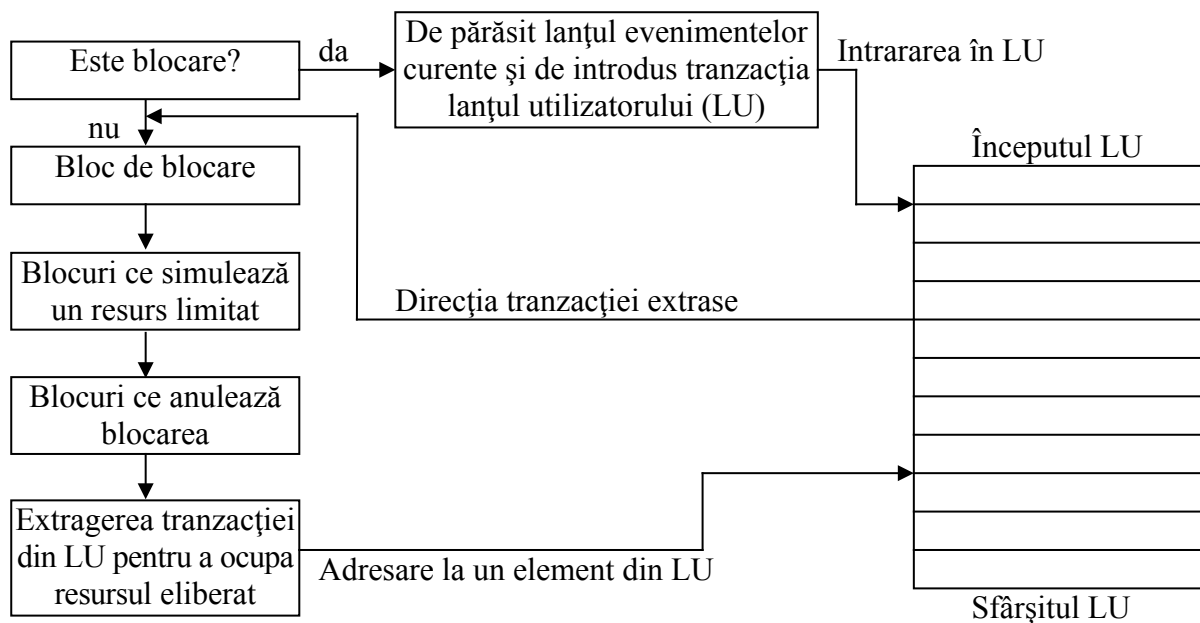
**MATCH**            **A**

$$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^{*n} \\ *n \\ Str \end{array} \right\}$$

### Lanțul utilizatorului

Lanțul utilizatorului permite de a extrage tranzații din model pentru un timp și de a le întoarce înapoi după dorință. Aceasta permite de a micșora lanțul evenimentelor curente și corespunzător de a micșora timpul de calcul. Într-un model pot fi câteva lanțuri ale utilizatorului.

Logica de funcționare:



1. Blocul LINK este destinat pentru extragerea tranzacției din lanțul evenimentelor curente și înregistrarea ei în lanțul utilizatorului. Câmpul A indică numele (numărul) lanțului utilizatorului, câmpul B arată ce algoritm este folosit pentru aranjarea tranzațiilor în lanț:
  - FIFO (First In – First Out) – tranzația este aranjată la sfârșitul lanțului. Extragerea din el se face de la începutul lui.
  - LIFO (Last In – First Out) – tranzația este introdusă la începutul lanțului.
  - $P_j$  – locul tranzației în lanț este determinat de parametrul  $P_j$ . În măsura micșorării valorii parametrului tranzația este introdusă mai aproape de începutul lanțului. În caz de egalitate a acestui parametru pentru două tranzații, ele se includ în lanț corespunzător cu timpul lor de sosire.

Câmpul C este o ieșire alternativă.

Dacă indicatorul de legătură cu lanțul este conectat, tranzația se va include în lanț, în caz contrar ea va trece în blocul indicat în câmpul C.

Formatul:

<b>LINK</b>	<b>A</b>	<b>B</b>	<b>C</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} FIFO \\ LIFO \\ P_j \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$

Ex. LINK \*1,LIFO ; a include tranzacția la începutul LU numărul căruia este scris în primul parametru.

2. Blocul UNLINK îndeplinește operația inversă blocului LINK. El este destinat pentru eliminarea tranzacțiilor din lanțul utilizatorului. Câmpul A arată numărul lanțului utilizatorului. Câmpul B indică la blocul, în care vor fi transferate tranzacțiile extrase din lanț. În câmpul C se organizează un contor al tranzacțiilor extrase. Înscierea ALL în acest câmp înseamnă extragerea tuturor tranzacțiilor. Câmpul D poate avea următoarele argumente:

2.1. Numărul unui parametru al tranzacției din lanțul utilizatorului.

În caz dacă câmpul E este vid, valoarea parametrului corespunzător a tranzacției care intră în blocul UNLINK se va compara cu parametrii corespunzători a tranzacțiilor din lanț. Dacă câmpul E nu este vid, atunci parametrii corespunzători ai tranzacțiilor din lanț se vor compara cu valoarea argumentului din câmpul E (comparare la neegalitate). Tranzacțiile, asupra cărora se îndeplinește condiția de comparare vor fi extrase din lanțul utilizatorului și vor fi transferate pe adresa din câmpul B.

2.2. Cuvântul BACK.

Din lanțul utilizatorului vor fi extrase un număr (indicat în câmpul C) de tranzacții, de la sfârșitul lanțului utilizatorului. Câmpul E trebuie să fie vid.

2.3. Variabilă booleană BV<sub>j</sub>.

Pentru fiecare tranzacție se calculează o variabilă booleană. Din lanțul utilizatorului vor fi extrase numai acele tranzacții, pentru care BV<sub>j</sub>=1. Dacă LU este vid (CH<sub>j</sub>=0) sau nu sunt tranzacții care satisfac condiția necesară, atunci tranzacția care intră în blocul UNLINK va încerca să treacă pe adresa indicată în câmpul F, dacă câmpul F este vid – în blocul următor.

Câmpul E conține un argument valoarea căruia se compară cu parametrul tranzacției(lor) din LU, indicat în câmpul B. Câmpul F indică la adresa alternativă, unde tranzacția va trece în caz dacă nu se îndeplinesc condițiile date. Se poate de evidențiat șase cazuri de complectare a câmpurilor acestui bloc.

- a) A,B,C – prezente; D,E,F – vide;
- b) A,B,C,F – prezente; D,E – vide;
- c) A,B,C,D – prezente; E,F – vide;
- d) A,B,C,D,F – prezente; E – vid;
- e) A,B,C,D,E – prezente; F – vid;
- f) A,B,C,D,E,F – prezente.

Formatul:

<b>UNLINK</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ ALL \\ BACK \\ BV_j \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$

Ex. SIMULATE

```

.....
GENERATE 100, FN$EXPON
QUEUE 1
LINK 1, FIFO, CAN
CAN SEIZE 1
DEPART 1
    
```

```

ADVANCE      700,FN$EXPON
RELEASE      1
UNLINK       1,CAN,1
TERMINATE    1
START        1000
END

```

### Grupuri

GPSS permite de a grupa obiectele sau atributele lor. Există două regimuri de grupare:

- 1) tranzacțional
- 2) numeric

În regimul tranzacțional se grupează tranzacțiile cu numerele lor corespunzătoare. În regim numeric grupul constă dintr-o listă de valori (ca exemplu: coeficienții de utilizare ai dispozitivelor care depășesc un nivel anumit).

În regim tranzacțional este posibilă schimbarea atributelor tranzacțiilor indiferent în ce loc al modelului ele se află. Dirijarea cu grupurile se face de către utilizator.

Grupurile au numai un ANS: C<sub>j</sub> – numărul de componente în grup. Sunt 5 blocuri care lucrează cu grupurile.

1. Blocul JOIN. Cu ajutorul acestui bloc, o tranzacție sau un număr se includ în grup (după disciplina FIFO).

a) Regimul tranzacțional. Dacă în blocul JOIN, este indicat numai câmpul A se consideră că acesta este regimul tranzacțional. Orice tranzacție care intră în blocul JOIN se include în grupul indicat în câmpul A. Regimul grupului este ales în considerație de primul obiect inclus în bloc. Tranzacția poate să se afle simultan în câteva grupuri.

Ex. JOIN 20 ;tranzacția va fi inclusă în grupul 20  
 JOIN PR ;tranzacția va fi inclusă în grupul care corespunde priorității lui.

b) Regimul numeric – se alege prin completarea câmpurilor A și B. În acest caz se calculează valoarea argumentului câmpului B, și valoarea numerică calculată se include în grupul cu numărul din câmpul A. Orice valoare numerică poate fi inclusă în grup doar numai o singură dată.

Ex. JOIN 20,7  
 JOIN ADA,V\$VAR

Formatul:

<b>JOIN</b>	<b>A</b>	<b>B</b>
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$

2. Blocul REMOVE permite de a extrage componente din grup. El de asemenea lucrează în două regimuri.

a) Regimul tranzacțional. În câmpul A se indică numărul grupului. Simulatorul verifică dacă tranzacția care intră în acest bloc aparține grupului dat și o extrage din grup dacă-i aparține. Dacă tranzacția nu aparține grupului dat, atunci el va fi transmis la adresa alternativă (câmpul F) sau la blocul următor (F este vid).

Ex. REMOVE 15  
 REMOVE GRUP,,,,,ALT

Blocul REMOVE poate să funcționeze în regimul de extragere din grup a unui număr dat de tranzacții, atributele cărora satisfac condițiile indicate. Pentru aceasta se folosesc câmpurile B,D,E.

În câmpul B se indică numărul de tranzacții care se extrag. Dacă câmpurile D și E nu sunt prezente, atunci din grup se extrage un număr indicat de tranzacții după disciplina FIFO. În câmpul D se indică argumentul după care vor fi extrase tranzacțiile din grup în caz de egalitate a lui cu argumentul din câmpul E. La indicarea ALL în câmpul A, vor fi extrase tranzacțiile care satisfac condiția dată. Dacă câmpul B este vid, el se tractează egal cu 1.

Ex. REMOVE 1,ALL  
 REMOVE 3,10,,PR,6



Dacă în câmpul D se va scrie un număr, el se va interpreta ca număr al parametrului.

b) Regimul numeric. În acest regim sunt folosite câmpurile A,C,F, unde A este numărul sau numele grupului, C – o valoare numerică și F – ieșire alternativă. Simulatorul verifică dacă există, în grupul cu numele din câmpul A, valoarea numerică, indicată în câmpul C. În caz de rezultat pozitiv această valoare se extrage din grup, iar în caz contrar, tranzacția trece la adresa alternativă sau în blocul următor.

Ex. REMOVE 5,,V\$NAME,,,ADR1

Formatul:

REMOVE	A	B	C	D	E	F
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ ALL \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ PR \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$

3. Blocul EXAMINE permite de a selecta direcția de mișcare a tranzacției în dependență dacă se află el sau o valoare dată în grup. Acest bloc lucrează în două regimuri.

a) Regimul tranzacțional. Dacă tranzacția care intră în grupul cu numele indicat în câmpul atunci ea trece la blocul următor. În caz contrar ea trece la adresa alternativă indicată în câmpul C. Câmpul B în acest regim nu se folosește.

Ex. EXAMINE 10,,ALT

b) Regimul numeric. În acest regim se îndeplinesc aceleași acțiuni ca și în cel tranzacțional cu excepția că obiectul de căutare este o valoare numerică, indicată în câmpul B.

Ex. EXAMINE 20,PR,ADR

Formatul:

EXAMINE	A	B	C
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$

4. Blocul SCAN funcționează numai în regim tranzacțional și permite utilizatorului de a avea acces la atributele tranzacțiilor oriunde nu s-ar afla ele. Acest bloc permite:

a) De a determina prezența între membrii grupului a tranzacțiilor cu valorile atributelor indicate.

b) De a primi valorile atributelor tranzacției care este membră a grupului, pentru care se îndeplinesc condițiile date.

c) De a schimba direcția de mișcare a tranzacțiilor în caz dacă în grup nu sunt tranzacții care satisfac condițiile indicate.

Câmpul A arată numărul grupului ce se precaută. În câmpul B se indică atributul de comparare (parametru sau prioritate). În câmpul C se indică argumentul comparării a atributului corespunzător. Dacă în grup se găsește o tranzacție, valoarea atributului căreia este egală cu argumentul câmpului C, atunci valoarea atributului din câmpul D a tranzacției (membru al grupei) se scrie în parametrul cu numărul indicat în câmpul E al tranzacției care intră în blocul SCAN. În caz de insucces al căutării, tranzacția va trece la adresa alternativă (câmpul F) sau blocul următor (F-vid).

Formatul:

SCAN	A	B	C	D	E	F
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ PR \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ PR \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$

Ex. SCAN 1,PR,6,,,ADR1 ;dacă în grupul 1 este o tranzacție cu PR=1,atunci tranzacția care intră va merge mai departe.

SCAN 4,10,X1,PR,6,NEXT ; pentru prima tranzacție din grupul 4 la care P10=X1; parametrului 6 al tranzacției care intră în bloc i se atribuie valoarea priorității a tranzacției găsite.

5. Blocul ALTER permite de a schimba atributele tranzacțiilor din grup. Acest bloc dă posibilitate:

- a) de a schimba valoarea atributului dat la toți membrii grupului;
- b) de schimbat atributul numai pentru o parte din membrii grupului;
- c) de schimbat valoarea atributului dat numai în cazul îndeplinirii condițiilor date.

În câmpul A se indică numărul grupului. În câmpul B – numărul de tranzacții din grup a căror atribut va fi schimbat (dacă în câmpul B este ALL pentru toate tranzacțiile). În câmpul C se indică atributul tranzacțiilor din grup care va fi schimbat. Orice număr din acest câmp se va interpreta ca număr al parametrului, iar PR – va modifica prioritatea. Câmpul D arată noua valoare a atributului. În câmpul se scrie atributul tranzacției (numărul se interpretează ca număr al parametrului, PR – prioritate). Înainte de a modifica valoarea atributelor membrilor grupului, valoarea acestui atribut (din câmpul E) se compară la egalitate cu atributul din câmpul F. Câmpul G indică la adresa alternativă pentru cazul când grupul este fără elemente sau nu sunt tranzacții care satisfac condițiile date.

Formatul:

ALTER	A	B	C	D	E	F	G
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ ALL \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ PR \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ PR \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$

Ex. ALTER 1,ALL,PR,100

ALTER 3,ALL,PR,0,1,10,ALT ; pentru tranzacțiile care au P1=10,

prioritatea se instalează în 0. Dacă nu sunt de acest fel (P1=10) se folosește de ieșirea alternativă ALT. prioritatea se instalează în 0. Dacă nu sunt de acest fel (P1=10) se folosește de ieșirea alternativă ALT.

### Obiecte ale categoriei de memorare. Celule de memorie.

Celulele sunt folosite pentru salvarea diferitor valori de numere și extragerea lor în statistica finală. Există 2 formate de celule: a) cuvânt –  $2^{31}-1$ ; b) semicuvânt –  $2^{15}-1$ .

ANS ale celulelor sunt  $X_j$  și  $XH_j$  ce întorc valoarea celulei respectiv pentru formatul cuvânt și semicuvânt.

1. Blocul SAVEVALUE permite de a scrie în celula, numărul sau numele căreia este indicat în câmpul A, a unei valori dorite (constantă sau ANS) ce se indică în câmpul B. Iar câmpul C determină formatul valorii (cuvânt sau semicuvânt). Implicit stă formatul cuvânt, dacă nu indicăm nimic, iar dacă dorim formatul semicuvânt se scrie în acest câmp simbolul "H". Dacă vom încerca să scriem o valoare de formatul cuvânt într-o celulă semicuvânt simulatorul ne va preîntâmpina. În acest caz octeții superiori se vor pierde și modelarea va continua. Blocul SAVEVALUE funcționează în trei regimuri: increment, decrement și înlocuire. Aceste regimuri sunt analogice regimurilor blocului ASSIGN. În cazul înlocuirii valoarea nouă indicată în câmpul B este înlocuită cu cea veche. Pentru increment (stă '+' după argumentul A), celulei îi se atribuie valoarea veche incrementată cu valoarea nouă din B. Și pentru decrement (stă '-' după argumentul A), celula i se atribuie valoarea veche decrementată cu argumentul din B.

Formatul:

SAVEVALUE	A	B	C
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	[H]

Ex. SAVEVALUE 1+,30

SAVEVALUE COEF,V\$COEF

### Matrice de celule

Matricele permit de a lega atribute adăugătoare la obiectele GPSS. Pentru folosirea matricelor ele trebuie mai întâi să fie declarate. Pentru aceasta se folosește cartela MATRIX.

Formatul cartelei:

<numeM> **MATRIX** A,B,C

unde A este formatul celulelor X sau H (cuvânt sau semicuvânt), B – numărul de linii, C – numărul de coloane.

Ex. ALFA MATRIX X,5,5

1. Blocul MSAVEVALUE funcționează analogic blocului SAVEVALUE, dar informația se va scrie în elementele matricei cu numele indicat în câmpul A, iar valoare modificată este indicată în câmpul D. Câmpurile B și Carată linia și coloana corespunzătoare la intersecția cărora se va scrie valoarea nouă a elementului. Câmpul E indică formatul elementului.

Formatul:

MSAVEVALUE	A	B	C	D	E
	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \end{array} \right\}$	[H]

Ex. MSAVEVALUE ALFA-,3,2,20

**Cartela INITIAL** permite de a atribui valori inițiale obiectelor din categoria de memorare și de asemenea de a atribui starea cheilor logice înainte de modelare. Implicit valoarea celulelor și matricelor până la modelare sunt egale cu 0. Dacă avem mai multe obiecte de inițializat ele se separă prin '/

Ex. INITIAL X10,13/LS1/MX(12,1),34

### Chei logice

Cheile logice sunt folosite pentru simularea obiectelor cu două stări. Orice cheie poate să se afle în una din două stări: a) setat; b) resetat (înaintea modelării toate sunt în această stare)

Cheile au două ANS: LR<sub>j</sub> și LS<sub>j</sub>. Starea cheilor se analizează de către blocul GATE ce permite de a schimba direcția de mișcare a tranzacțiilor în dependență de starea cheii corespunzătoare.

1. Blocul LOGIC instalează cheia logică într-o stare dorită. La intrarea tranzacției în acest bloc, starea cheii indicate în câmpul A poate fi modificată în dependență de mnemonica care urmează după acest bloc, în felul următor:

S – cheia se setează

R – cheia se resetează

I – cheia se inversează.

Formatul:

LOGIC mnem	A
$\left[ \begin{array}{l} S \\ R \\ I \end{array} \right]$	$\left\{ \begin{array}{l} C \\ ANS_j \\ ANS^*n \\ *n \\ Str \end{array} \right\}$

Ex. LOGIC R 10  
LOGIC I X\$BETA

Pentru instalarea înainte de modelare în stare setată se folosește cartela INITIAL.

Ex. INITIAL LS1/LS\$ALFA/LS15